Designing Constructive Machine Learning Models based on Generalized Linear Learning Techniques

Parisa Kordjamshidi
Department of Computer Science
KU Leuven, Belgium
parisa.kordamshidi@cs.kuleuven.be

Marie-Francien Moens
Department of Computer Science
KU Leuven, Belgium
sien.moens@cs..kuleuven.be

Abstract

We propose a general framework for designing machine learning models that deal with constructing complex structures in the output space. The goal is to provide an abstraction layer to easily represent and design *constructive learning models*. The learning approach is based on generalized linear training techniques, and exploits techniques from combinatorial optimization to deal with the complexity of the underlying inference required in this type of models. This approach also allows to consider global structural characteristics and constraints over the output elements in an efficient training and prediction setting. The use case focuses on building spatial meaning representations from text to instantiate a virtual world.

1 Introduction

Designing learning models for real world problems that construct complex output structures containing various elements, for instance, entities, relationships and their attributes is very challenging. Punyakanok et al. [10] describe three fundamentally different and high level solutions for structured output prediction: **Learning only (LO):** Local classifiers are trained and used to predict each output component separately. **Learning plus inference (L+I):** Training is performed locally as in the LO models, but the global constraints/correlations among components are imposed during prediction [3]. **Inference based training (IBT):** Inference is used during training so that the constraints and dependencies among the variables are incorporated into the training process.

However for training, there is a spectrum of various model compositions between two extreme sides of only local training as in LO and L+I schemes versus a full global training in the IBT scheme [11]. The structural dependencies that are considered in the learning models are not always according to the dependencies represented in the data model in the relational domains [9]. This can be due to the resulting computational complexities or because the relational models do not represent the dependencies between the features of the entities. Therefore having an expressive representation of the learning model in addition to the data model is always useful and eases designing, assessing, decomposing and improving the learning models.

In this paper, we provide a simple abstraction for designing global structured learning models (i.e. IBT) for domains that construct models based on entities, relationships and their attributes. Particularly, we focus on natural language meaning representations obtained by semantic parsing of text. The meaning representations regard the objects or persons and their spatial relationships in a described scene. We provide a generic approach for representing input and output spaces in a relational domain. We integrate our framework in the non-probabilistic structured output prediction models. We explain the way we build the objective functions for the inference during training and during prediction according to the relational input/output data and the knowledge about the structural characteristics of the output in a framework which we name *Link-And-Label* (LAL) model. In the LAL model we do collective classification of entities and their relationships in a structured learn-

ing framework compared to ad hoc collective approaches. In this way the theoretical guarantees for the generalization bounds are hold although we exploit approximate inference [4]. Our application target is to instantiate or create a virtual reality based on the information that is found in a text.

2 Structured output learning

We work in a supervised structured learning setting which is briefly described here. In the *supervised* setting one learns a mapping $h : \mathcal{X} \rightarrow \mathcal{Y}$ between the input space $\mathcal{X}$ and discrete output space $\mathcal{Y}$ given a set of examples, $E = \{(x^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y} : i = 1 \dots N\}$. In the *structured* learning, given the complex inputs and outputs, we learn a $g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ over input-output pairs. Then for prediction, we need an inference over g to find the best y for a given x. Thus h is, $h(x; W) = \arg\max_{y \in \mathcal{Y}} g(x, y; W)$. The function g is assumed to be linear over a combination of input and output features $f(x, y)$ i.e. $g(x, y; W) = \langle W, f(x, y) \rangle$ [16]. A popular discriminative training method is to minimize the below convex upper bound of the loss function over the training data:

$$l(W) = \sum_{i=1}^{N} \max_{y \in \mathcal{Y}} (g(x^i, y; W) - g(x^i, y^i; W) + \Delta(y^i, y)), \tag{1}$$

the inner maximization is called loss-augmented inference and finds the most violated output per training example. This is a crucial inference problem to be solved during training of such models.

3 Link-And-Label model

The Link-And-Label name is inspired by the conceptualization process that a human does when extracting pieces of information from arbitrary inputs, and trying to connect them to some concepts which might be of interest in the output. We usually group objects or link them to each other and label the groups with more abstract concepts. For example in text understanding, the various segments of the text are linked to each other and are labeled as an instance, or an indicator of a specific object (such as a trajector or a landmark when considering spatial meaning). These labels are the new properties of the higher level concepts. Again by linking a number of labeled objects (for example, in the case of a composed-of relationship) we build more complex concepts and tag them with new labels indicating the type of the relationships or their attributes (e.g. indicating a spatial relation which itself can be an overlapping relationship or disconnected relationship). Concepts can have relationships of different types, which are usually defined in a domain ontology. The relationships between concepts describe the relationships between the instances of them. This property stimulates to design and represent a learning model that exploits a first order representation in terms of input component types and output label types. In such a setting we can easily restrict the type of input component to a certain type of output label and use this representation in the learning objective, which is explained more in detail below. To explain the Link-And-Label model, first we describe the terminology that we use based on the *input* and *output* distinction. Then we describe the form of the objective function of the training and the prediction for constructive learning in this framework.

3.1 Input and output spaces

Each input x is a set of components $\{x_1..x_K\}$. Each component has a *type*. Each $x_k \in x$ is described by a vector of features relevant for its type. The feature vector is denoted by ϕ_p where p is an index that refers to a specific type. For instance, in semantic labeling of text an input type can be a word (atomic component) or a pair of words (composed component), and each type is described by its own features (e.g. a single word by its part-of-speech, the pair by the distance of the two words). The features that describe a property of an atomic component are called local and the ones that describe the relation between more than one atomic component are called relational features.
The output space y is represented by a set of *labels* $\mathbf{l} = \{l_1, .., l_P\}$. The labels are defined based on the elements in the output and can have semantic relationships to each other. To be able to represent complex output concepts in general for any arbitrary task, we distinguish between two types of labels, the *single labels* and *linked labels* that refer to an independent concept and to a configuration of a number of related single labels respectively. Linked labels can represent different types of semantic relationships between single labels. They can express *composed-of*, *is-a* and other semantics. For convenience, to show which labels are connected by a *linked label*, we represent

the *linked labels* by a *label string*. This is a concatenation of the labels that are linked together and construct a bigger semantic part of the whole output. For example a *spatial relation* can be denoted by $sp.tr.lm$ meaning that it is *composed of* the three single labels, sp (spatial indicator), tr (trajector) and lm (landmark). A label string only shows the links between the labels and the semantics of the links should be clarified and given to the learning model later. The semantics can be defined by means of some basic rules or any complex grammar.

3.2 Connecting input and output spaces

Labels are binary indicators that receive an input component and indicate whether it has that certain label. This is similar to generating a joint feature function for the case of multi-class classification [16] which we generalize for the relational case and for arbitrary output structures. The binary indicator function for each linked label is defined according to its semantics. For example, a spatial relation label is defined in a way to convey the *composed-of* semantics based on the labels of its components. We use both notations of $l_p(x_k)$ or shorter l_{pk} to indicate the membership of the component x_k in the set of components with label l_p. To formally specify the connections between input components and output labels we use the notion of *template* as in relational graphical models [12, 14, 2]. The learning model is specified with a set of templates $\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_P\}$. Each template $\mathcal{C}_p \in \mathcal{C}$ is specified by three main characteristics,
(1) *A subset of joint features*. This is referred to as local joint feature function and defined over a number of input type(s) and output label(s) associated to the template $\mathcal{C}_p$. It is denoted by $f_p(x_k, l_p)$, where x_k is an input component, and l_p is a single label/linked label.
(2) *Candidate generator*. It generates candidate components upon which the specified subset of joint features is applicable, the set of candidates for each template is denoted as C_{l_p}.
(3) *A block of weights* W_p. This is a block of the main weight vector W of the model which is associated to that template and its local joint feature function.
LAL objective function. The main objective *discriminant function* $g = \langle W, f(x, y)\rangle$, is a linear function in terms of the combined feature representation associated to each candidate input component and an output label according to the template specifications. Given the design of the model using the templates the structure of the objective becomes transparent. The objective is written in terms of the instantiations of the templates and their related blocks of weights W_p in $W = [W_1, W_2, .., W_P]$,

$$g(x, y; W) = \sum_{l_p \in \mathbf{l}} \sum_{x_k \in C_{l_p}} \langle W_p, f_p(x_k, l_p)\rangle = \sum_{l_p \in \mathbf{l}} \sum_{x_k \in C_{l_p}} \langle W_p, \phi_p(x_k)\rangle l_{pk} = \sum_{l_p \in \mathbf{l}} \langle W_p, \sum_{x_k \in C_{l_p}} (\phi_p(x_k) l_{pk})\rangle \tag{2}$$

where the local joint feature vector $f_p(x_k, l_p)$, is an instantiation of the template $\mathcal{C}_{l_p}$ for candidate x_k. This feature vector is computed by scalar multiplication of the input feature vector of x_k (i.e. $\phi_p(x_k)$), and the output label l_{pk}. This output label is the indicator function of label l_p for component x_k. Each indicator function of a template linked label is applied on the relevant input component and its value is one when the intended semantics behind it holds for that component. For example, if a template is intended to do a logical *and* over the constituent labels, it means the indicator function of the linked label is one if all included single label indicators are one when applied on the input parts. Given this objective function, we can view the inference task as a *combinatorial constrained optimization* given the polynomial g which is represented in terms of labels, subject to the constraints that describe the relationships between the labels. For example, the *is-a* relationships can be defined as the following constraint, $(l(x_c) = 1) \Rightarrow (l'(x_c) = 1)$, where l and l' are two distinct labels that are applicable on the components with the same type of x_c. This problem can be solved using linear programming relaxations and in many cases just by using off-the-shelf solvers.

In general for designing such models, the highly correlated labels should ideally be linked to each other and be considered in one template. However, considering the global correlations in one template can become very complex. Hence, these global correlations are modeled via adding constraints [3]. The constraints can hold between the instantiations of one template which implies the relations between the components of one type also referred to as *autocorrelations* as defined in a *relational dependency networks* [9]. The constraints are exploited during training in the loss-augmented inference and are imposed on the output structure during prediction. We treat the objective as a linear function in which the association between labels in addition to their global relationships are modeled via linear constraints. For inference over an input example, we build a new instance of the objective function and *propositionalize* the first order constraints.

3.3 Component based loss function

We define the loss function (Δ) that is decomposable in the same way as the joint feature function. This is to avoid increasing the complexity of the *loss-augmented* inference compared to the prediction time inference [13]. We define a component-based loss for each template label l_p by measuring the Hamming loss between the vector of predicted labels for all candidates (denoted by Λ_{l_p}) and the ground truth assignments (denoted by Λ'_{l_p}) and normalize by the number of candidates,

$$\Delta_{l_p}(\Lambda, \Lambda') = \frac{1}{|C_{l_p}|} \sum_{k=1}^{|C_{l_p}|} \Delta_H(l_{pk}, l'_{pk}) \text{ where } \Delta_H(l_{pk}, l'_{pk}) = l_{pk} + l'_{pk} - 2l_{pk}l'_{pk}. \quad (3)$$

In this way we perform collective classification of input components and jointly minimize the loss for all label assignments. The labels can be weighted based on their importance in the output:

$$\Delta(y, y') = \sum_{p=1}^{P} \omega_{l_p} \Delta_{l_p}(\Lambda_{l_p}, \Lambda'_{l_p}), \quad (4)$$

ω_{l_p} is the weight of each label l_p and P is the number of templates. This linear loss in terms of the labels, provides a similar objective for training and prediction (in terms of variables and constraints).

4 Communicative inference

Solving the objective function in Equation 2 augmented by the loss function in Equation 4, during training can become highly inefficient for many relational data domains. This is because linearizing the linked labels in the objective function and the propositionaliztion of the constraints often produces a large number of output labels and constraints per training example. To deal with this problem we propose an additional layer of decomposition as a meta frame for applying off-the-shelf LP solvers. We propose an approach for decomposing the prediction and training time inference, which we name *communicative inference*. The basic idea is that given a *decomposition* by an expert, the inference *subproblems* are solved independently but they communicate to each other by *passing messages*, that is, passing solutions. To implement this idea we use a kind of block coordinate decent (BCD) [15] also referred to as alternating optimization (AO) [1]. In these methods, given a general objective function H of multivariate y, to find the MAP (Maximum a Posteriori) of H we can divide the variables into a number of blocks assuming that each block has a local maximizer. This approach can be used during training as well as prediction. This meta frame is to decompose a complex objective function at the problem layer according to the semantics of the problem as in our application instead of at the solution layer where the solvers try to provide efficient solutions.

5 Experimental results

The proposed model is evaluated on the spatial role labeling data offered during SemEval 2012 [5], containing 12013 sentences which considers recognizing spatial objects (i.e. trajectors, landmarks and spatial-indicators) and their spatial relations (triplets) [8]. We have extended this with qualitative spatial attributes here that describe the type of the spatial relationships in terms of formal spatial representation models including the directional (left, right...), regional (externally connected, disconnected, ...) and distal [7, 6]. These attributes construct a lightweight ontology where nodes in the ontology have composed-of or is-a relationships to each other. There are some other properties such as mutual exclusivity between the semantic labels, etc.We implement a number of models using structured support vector machines (SSVM) and averaged structured perceptrons (AvgSP). The experimental results show that using global structural characteristics of the spatial language and the spatial ontology in the form of constraints during training and prediction improves the results compared to training local classifiers (using 10-fold cross validation). First, this improvement was from F1=0.49 by local binary classifiers to F1=0.52 for spatial triplets when we did global inference during prediction for spatial roles and relations. The results improved to F1=0.579 when we did global inference during training (by SSVM). The results were better when using AvgSP. The global training and prediction by AvgSP provided F1=602. Second, we trained a global model for prediction of the multiple attributes of the spatial relations and added a layer to the spatial role and relation extraction. Pipelining the two global models provided a micro averaged F1=0.526 for the predicted attributes of

spatial relations. We could train a global model encompassing all roles, relations and their attributes using our proposed *communicative inference* during *training* and *prediction*. These results improved to F1=0.605 for relations and micro averaged F1=0.529 for attributes when we used communicative inference only during *prediction* for the two global models (considering hierarchical relationships and mutual exclusivity constraints imposed on the predicted labels). When the communicative inference used during *training* the results of the spatial relation extraction layer improved to F1=0.617 but there was a drop in the prediction of the attributes to micro averaged F1=0.50 (by AvgSP).

6 Conclusions

The proposed link-and-label model for constructive learning can be viewed as a general framework for collective classification in the frame of structured output prediction in relational data domains. Particularly, we consider natural language meaning representations in terms of concepts (entities), their links (relationships) and the attributes. The proposed model is based on generalized linear models and benefits from established theoretical guarantees of structured output prediction. We use the notion of templates as in relational graphical models to represent the dependencies among input and output components of the learning. The component based loss which we use is decomposable as the feature function and makes our model sufficiently general in learning with any arbitrary structure and exploiting ontological constraints. Combinatorial optimization and decomposition techniques make the inference during training and prediction tractable. The application contributes to the task of constructing spatial meaning representations from text to instantiate a virtual world.

Acknowledgements

This research was funded by the DBOF/08/043 grant from KULeuven and the MUSE project (EU FP7-296703).

References

[1] J. C. Bezdek and R. Hathaway. Some notes on alternating optimization. In Nikhil R. Pal and Michio Sugeno, editors, *Advances in Soft Computing*, volume 2275 of *LNCS*, pages 288–300. 2002.

[2] R. Bunescu and R. J. Mooney. Statistical relational learning for natural language information extraction. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, pages 535–552. MIT Press, 2007.

[3] M. W. Chang, L. A. Ratinov, and D. Roth. Structured learning with constrained conditional models. *Machine Learning*, 88(3):399–431, 2012.

[4] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *(ICML)*, pages 304–311. ACM, 2008.

[5] P. Kordjamshidi, S. Bethard, and M. F. Moens. SemEval-2012 task 3: Spatial role labeling. In *Proceedings of (SemEval-2012)*, volume 2, pages 365–373. ACL, 2012.

[6] P. Kordjamshidi, M. van Otterlo, and M. F. Moens. From language towards formal spatial calculi. In Robert J. Ross, Joana Hois, and John Kelleher, editors, *In (CoSLI'10, at Spatial Cognition)*.

[7] P. Kordjamshidi, M. van Otterlo, and M. F. Moens. Spatial role labeling: task definition and annotation scheme. In *Proceedings of (LREC'10)*, pages 413–420, 2010.

[8] P. Kordjamshidi, M. van Otterlo, and M. F. Moens. Spatial role labeling: towards extraction of spatial relations from natural language. *ACM - Transactions on Speech and Language Processing*, 8:1–36, 2011.

[9] J. Neville and D. Jensen. Relational dependency networks. *JMLR*, 8:653–692, 2007.

[10] Vasin Punyakanok, Dan Roth, Wen Tau Yih, and Dav Zimak. Learning and inference over constrained output. In *Proceedings of IJCAI'05*, pages 1124–1129. Morgan Kaufmann Publishers Inc., 2005.

[11] R. Samdani and D. Roth. Efficient decomposed learning for structured prediction. In *ICML*, 2012.

[12] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of UAI'02*, pages 485–492. Morgan Kaufmann Publishers Inc., 2002.

[13] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Proceedings of NIPS*. MIT Press, 2004.

[14] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Proceedings of NIPS*. MIT Press, 2004.

[15] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

[16] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6(2):1453–1484, 2006.